ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

LÊ PHAN HỮU NGHĨA
NGUYỄN HOÀNG TUẤN
NGUYỄN VĂN DŨNG

ĐỒ ÁN CHUYÊN NGÀNH

PHÁT HIỆN LỆNH LIVING-OFF-THE-LAND BẰNG CÁCH SỬ DỤNG HỌC TÍCH CỰC

LIVING-OFF-THE-LAND COMMAND DETECTION USING ACTIVE LEARNING

CỬ NHÂN NGÀNH AN TOÀN THÔNG TIN

TP. Hồ Chí Minh, 2023

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

LÊ PHAN HỮU NGHĨA - 20520650
NGUYỄN HOÀNG TUẤN - 20522118
NGUYỄN VĂN DŨNG - 18520639

ĐỒ ÁN CHUYÊN NGÀNH

PHÁT HIỆN LỆNH LIVING-OFF-THE-LAND BẰNG CÁCH SỬ DỤNG HỌC TÍCH CỰC

LIVING-OFF-THE-LAND COMMAND DETECTION USING ACTIVE LEARNING

CỬ NHÂN NGÀNH AN TOÀN THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN:
ThS. Phan Thế Duy
ThS. Nghi Hoàng Khoa

TP.Hồ Chí Minh - 2023

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án chuyên ngành, chúng em xin gửi lời cảm ơn sâu sắc tới thầy Phan Thế Duy và thầy Nghi Hoàng Khoa, những người đã hướng dẫn và giúp đỡ chúng em trong suốt quá trình nghiên cứu và thực hiện đồ án.

Cả hai thầy tuy bận nhưng vẫn dành thời gian, kiến thức và kinh nghiệm của mình để hướng dẫn chúng em từ giai đoạn lựa chọn đề tài, xác định phương pháp nghiên cứu, đến việc thực hiện và trình bày kết quả. Nhờ sự chỉ bảo và hỗ trợ tận tình từ hai thầy, chúng em đã có cơ hội hiểu sâu hơn về lĩnh vực chuyên ngành và phát triển kỹ năng nghiên cứu, phân tích và giải quyết vấn đề.

Trân trọng.

Lê Phan Hữu Nghĩa **Nguyễn Hoàng Tuấn** **Nguyễn Văn Dũng**

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

TÓM TẮT ĐỒ ÁN

Tính cấp thiết của đề tài nghiên cứu:

Cuộc tấn công "Living-off-the-land" ngày càng trở nên nguy hiểm và làm gia tăng nguy cơ bị khai thác các hệ thống mạng. Bởi vì cuộc tấn công này sử dụng các dòng lệnh và công cụ hợp pháp có sẵn trong hệ thống, nó khó bị phát hiện và giúp kẻ tấn công tiếp cận và kiểm soát hệ thống của máy tính bị tấn công một cách rất tinh vi. Điều đáng lo ngại hơn, các lệnh này thường không bị các công cụ phòng thủ thông thường phát hiện, làm cho việc phòng chống trở nên khó khăn.

Đề tài "Living-off-the-land Command Detection using Active Learning" đóng vai trò quan trọng trong việc giải quyết vấn đề ngày càng tăng về cuộc tấn công "Living-off-the-land". Đề tài này của chúng em tập trung vào việc xây dựng một phương pháp phát hiện hiệu quả nhờ sử dụng phương pháp học chủ động để phân loại và ngăn chặn các lệnh độc hại được thực thi từ các công cụ hợp pháp trong hệ thống mạng. Cách tiếp cận này của tụi em mang lại lợi ích lớn trong việc cung cấp giải pháp mạnh mẽ để tăng cường an ninh mạng và đối phó với cuộc tấn công tinh vi.

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu vấn đề

Đề tài "Living-off-the-Land Command Detection Using Active Learning" là một vấn đề quan trọng trong lĩnh vực an ninh mạng. "Living-off-the-Land" (LOTL) là thuật ngữ được sử dụng để chỉ các cuộc tấn công mà kẻ tấn công tận dụng các công cụ, ứng dụng và tài nguyên hệ thống có sẵn trên mạng để thực hiện các hoạt động độc hại mà không cần phải đưa vào hệ thống một tập tin độc hại.

Trong đồ án này của tụi em, việc phát hiện cuộc tấn công LOTL dựa trên việc sử dụng phương pháp học chủ động (active learning) là một phương pháp tiếp cận đáng chú ý. Học chủ động là một kỹ thuật học máy mà thu thập thông tin từ người dùng hoặc hệ thống để cải thiện hiệu suất của mô hình. Nó cho phép hệ thống tương tác với người dùng để xác minh các mẫu dữ liệu bất thường hoặc độc hại.

Vấn đề quan trọng trong cuộc tấn công LOTL là các lệnh (command) được sử dụng bởi kẻ tấn công để thực hiện các hoạt động độc hại. Tuy nhiên, các lệnh này thường được thiết kế để trông giống như các lệnh hợp lệ hoặc được thực hiện bởi các công cụ hệ thống thông thường, điều này khiến việc phát hiện trở nên khó khăn.

Phương pháp sử dụng học chủ động trong việc phát hiện cuộc tấn công LOTL tập trung vào việc xác định các lệnh bất thường và độc hại thông qua việc ghi nhận và phân tích các hoạt động của người dùng. Hệ thống sẽ theo dõi và thu thập dữ liệu về các lệnh được thực hiện trên hệ thống, sau đó sử dụng kỹ thuật học chủ động để yêu cầu phản hồi từ người dùng hoặc các chuyên gia an ninh để xác minh tính hợp lệ của các lệnh.

Việc sử dụng phương pháp học chủ động giúp cải thiện khả năng phát hiện và ngăn chặn các cuộc tấn công LOTL bằng cách học từ dữ liệu thực tế và kinh nghiệm của người dùng. Bằng cách liên tục cập nhật và điều chỉnh mô hình dựa trên phản hồi từ người dùng, hệ thống có thể nâng cao khả năng phát hiện các lệnh độc hại và giảm tỷ lệ các lệnh giả mạo.

Tóm lại, cuộc tấn công LOTL là một vấn đề nguy hiểm trong an ninh mạng và việc sử dụng phương pháp học chủ động có thể cung cấp một phương pháp hiệu quả để phát hiện và ngăn chặn các cuộc tấn công này bằng cách tận dụng thông tin từ người dùng và chuyên gia an ninh.Trong đồ án này tụi em tự tìm nguồn dữ liệu data.

1.2. Kiến thức nền tảng

Để hiểu vấn đề về cuộc tấn công "Living-off-the-land", cần có một số kiến thức nền tảng về các khái niệm trong lĩnh vực an ninh mạng và học máy. Dưới đây là một số khái niệm cơ bản có thể giúp ta hiểu rõ hơn về vấn đề này:

- An ninh mạng: An ninh mạng là lĩnh vực nghiên cứu và thực thi các biện pháp để bảo vệ hệ thống và dữ liệu trên mạng khỏi các cuộc tấn công, truy cập trái phép, và sự xâm nhập.

- Living-off-the-land (LOTL): là thuật ngữ để chỉ các cuộc tấn công mà kẻ tấn công tận dụng các công cụ, ứng dụng, và tài nguyên hệ thống có sẵn trên mạng để thực hiện các hoạt động độc hại mà không cần phải đưa vào hệ thống mã độc mới.

- Dòng lệnh (Command) và Command Detection: Trong đồ án của chúng em, "command" đề cập đến các lệnh hoặc câu lệnh được thực thi trên hệ thống để thực hiện các hoạt động như điều khiển, thực thi mã, truy cập dữ liệu, và thực hiện các tác vụ hệ thống khác. Command detection là quá trình xác định và phân tích các lệnh được thực hiện để phát hiện các lệnh độc

hại hoặc không hợp lệ.

- Học chủ động (Active Learning): là một phương pháp trong học máy mà mô hình học máy có khả năng tương tác với người dùng hoặc các chuyên gia để yêu cầu phản hồi hoặc xác minh thông tin từ dữ liệu. Phương pháp này giúp cải thiện hiệu suất và độ chính xác của mô hình bằng cách sử dụng thông tin từ người dùng để điều chỉnh quá trình học.

- Học máy (Machine Learning):là lĩnh vực nghiên cứu và ứng dụng của trí tuệ nhân tạo mà các mô hình và thuật toán được sử dụng để giúp máy tính học từ dữ liệu và tự động cải thiện hiệu suất hoặc dự đoán. Trong đồ án của chúng em, học máy được sử dụng để phát hiện và phân loại các lệnh LOTL dựa trên dữ liệu và phản hồi từ người dùng.

CHƯƠNG 2. KIẾN TRÚC VÀ CÁC KỸ THUẬT

2.1. Phương pháp

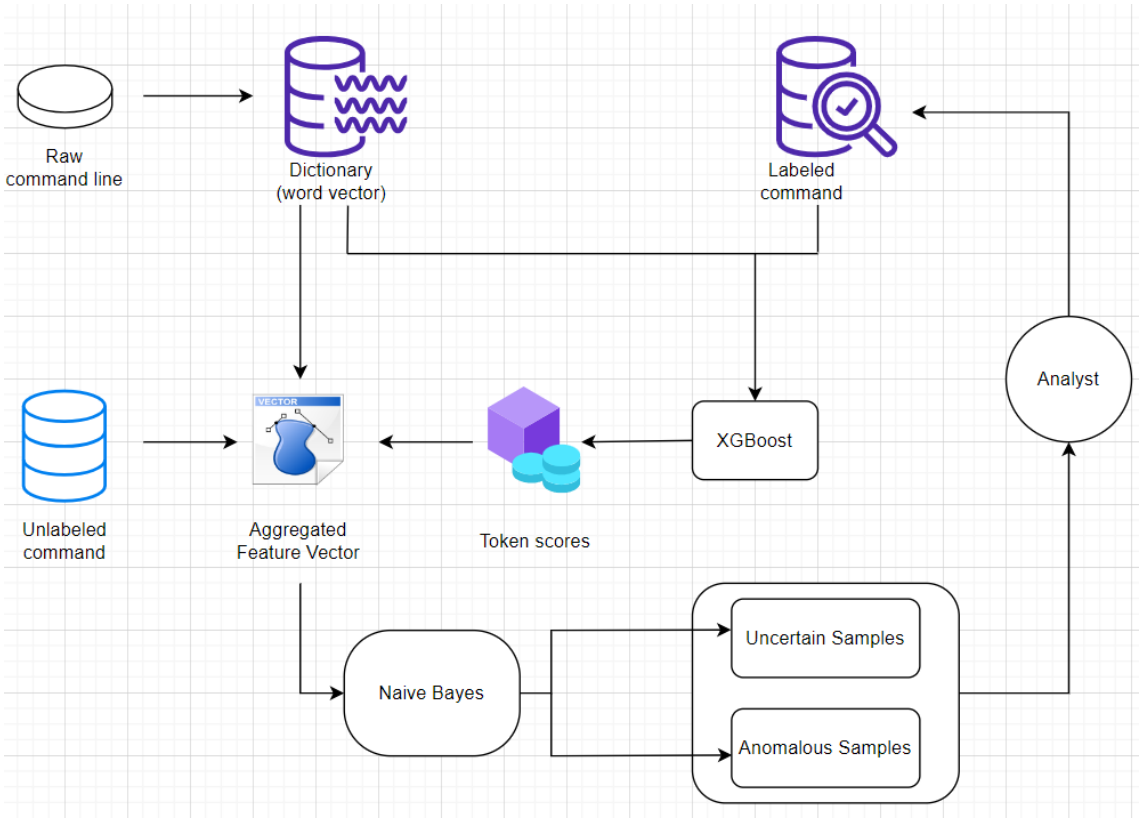
Kỹ thuật tính năng (Feature Engineering): phương pháp này tập trung vào việc chọn và trích xuất các đặc trưng quan trọng từ dữ liệu lệnh. Việc phân tích cú pháp, phân tích từ ngữ, trích xuất thông tin về quyền truy cập và các phân tích khác có thể được áp dụng để tạo ra các đặc trưng mô tả các lệnh.

Học chủ động (Active Learning): là phương pháp chính được sử dụng để thu thập thông tin từ người dùng hoặc chuyên gia để cải thiện hiệu suất của mô hình. Nó cho phép mô hình yêu cầu phản hồi để xác minh tính hợp lệ của các lệnh độc hại và không độc hại.

2.2. Kiến trúc

Tóm tắt sơ lược về kiến trúc: Mô hình sẽ khởi đầu với ba bộ dữ liệu là *Labeled command*, *Raw command* và *Unlabeled command* (dữ liệu cần kiểm tra). Tiếp theo đó, qua các quá trình xử lý, dữ liệu đầu tiên sẽ được huấn luyện cho mô hình XGBoost [1] để có thể tính điểm token - một thành phần quan trọng để tạo vector đặc trưng. Sau khi tạo vector đặc trưng với các nhãn tương ứng, ta sẽ huấn luyện nó qua mô hình Naive Bayes, Sau cùng mô hình vừa được huấn luyện đó sẽ được sử dụng để tính điểm không chắc chắn (Uncertainty) và điểm bất thường (Anomaly) để chọn ra các mẫu bất thường và các mẫu không chắc chắn cho các chuyên gia để phân tích.

Lưu ý: Dữ liệu cần kiểm tra chỉ sử dụng sau khi cả hai mô hình đều đã được huấn luyện.



Hình 2.1: *Kiến trúc thực nghiệm*

2.3. Các kỹ thuật

2.3.1. Các mô hình phân loại sử dụng trong quá trình thực hiện

- Random Forest: Random Forest [2] là một mô hình phân loại dựa trên quyết định từ nhiều cây quyết định (decision tree). Nó xây dựng nhiều cây quyết định độc lập nhau và sử dụng phương pháp bầu chọn (voting) để đưa ra dự đoán. Mỗi cây quyết định trong Random Forest được huấn luyện trên một tập con của dữ liệu và một tập con của các đặc trưng.

- SVM (Support Vector Machine): SVM [3] là một mô hình phân loại dựa trên máy vector hỗ trợ. Nó xây dựng một đường ranh giới tuyến tính hoặc phi tuyến tính để phân tách các lớp dữ liệu. SVM tìm kiếm đường ranh giới tối ưu dựa trên việc tối đa hóa khoảng cách từ các điểm dữ liệu gần nhất đến đường ranh giới.

- Light GBM (Light Gradient Boosting Machine): Light GBM [4] là một mô hình gradient boosting được sử dụng cho bài toán phân loại. Nó sử dụng thuật toán gradient boosting để xây dựng một tập hợp các cây quyết định nhằm tăng cường hiệu suất của mô hình. Light GBM có thể xử lý các tập dữ liệu lớn và tối ưu hóa hiệu suất thông qua việc sử dụng các kỹ thuật như phân hoạch theo lượng thông tin (information gain) và phân hoạch theo giá trị tốt nhất (best split value).

- Naive Bayes: Naive Bayes [5] là một thuật toán phân loại dựa trên nguyên tắc xác suất Bayes. Nó giả định rằng các đặc trưng độc lập với nhau và tính xác suất của một điểm dữ liệu thuộc vào từng lớp dựa trên xác suất của các đặc trưng.

- Kỹ thuật Ensemble Learning kết hợp các mô hình Random Forest, SVC (Support Vector Classifier) và Light GBM:

- Kỹ thuật XGBoost (Extreme Gradient Boosting) là một phương pháp Ensemble Learning dựa trên Boosting, sử dụng cây quyết định như mô hình con.

2.3.2. Các kỹ thuật xử lý dữ liệu

Min-pooling, Max-pooling và Avg-pooling:

- Pooling (còn được gọi là phép gộp) là một kỹ thuật thường được sử dụng trong mạng neural tích chập (CNN) để giảm kích thước của bản đồ đặc trưng.

- Max pooling (gộp tối đa) lấy giá trị lớn nhất trong mỗi cửa sổ pool và sử dụng nó làm giá trị đại diện cho cửa sổ đó.

- Min pooling (gộp tối thiểu) tương tự như max pooling, nhưng lấy giá trị nhỏ nhất trong mỗi cửa sổ pool.

- Avg pooling (gộp trung bình) tính giá trị trung bình của mỗi cửa sổ pool và sử dụng nó làm giá trị đại diện.

One-Hot Encoding (mã hóa one-hot):

- Mã hóa one-hot là một phương pháp để biểu diễn các biến phân loại dưới dạng vector nhị phân.Trong mã hóa này, mỗi giá trị duy nhất của biến phân loại được đại diện bởi một vector có độ dài bằng số lượng giá trị duy nhất.

Phân đoạn (Tokenization) và Kỹ thuật nhúng từ (Word embedding):

- Phân đoạn (Tokenization) là quá trình chia câu hoặc đoạn văn thành các phần tử nhỏ hơn gọi là "token". Các token có thể là từ, cụm từ, hoặc ký tự, phụ thuộc vào cách xử lý.

- Kỹ thuật nhúng từ (word embedding) là một quá trình biểu diễn từ trong không gian vector số.

Mục đích của kỹ thuật nhúng từ là biểu diễn từng từ dưới dạng một vector có đặc trưng, cho phép mô hình học được thông tin ngữ nghĩa và liên quan giữa các từ. Trong đề tài này, nhóm em dùng kỹ thuật nhúng từ Fasttext, và ngoài ra có tìm hiểu và thực hiện thêm về Word2Vec, GloVe.

Điểm cao nhất (Top scores):

- Top scores có thể được hiểu là "những điểm số cao nhất" hoặc "những điểm số hàng đầu". Thuật ngữ này thường được sử dụng để chỉ đến những điểm số cao nhất trong một danh sách hoặc bảng điểm. Điểm số hàng đầu thường được xem là những điểm số đạt được cao nhất hoặc ưu việt hơn so với các điểm số khác.

Số lượng mã (Number token):

- Number token trong ngữ cảnh này, nghĩa là số lượng token có trong mỗi dòng lệnh

Mã hiếm (Rare token):

- Rare token là một loại token được sử dụng để đại diện cho các từ hiếm hoặc không phổ biến trong ngữ liệu.

- Khi xử lý ngôn ngữ tự nhiên, các từ hiếm hoặc không phổ biến thường không có đủ thông tin để tạo ra một nhúng từ (word embedding) hiệu quả.Thay vào đó, các từ này được thay thế bằng một mã hiếm (rare token), tức là một token duy nhất dùng để đại diện cho tất cả các từ hiếm trong ngữ liệu.

- Mã hiếm thường được sử dụng để giảm kích thước của từ vựng và tăng tốc độ xử lý trong các mô hình xử lý ngôn ngữ tự nhiên.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Ở chương này chúng em sẽ trình bày các ràng buộc, yêu cầu của bài báo, sau đó đưa ra kết quả và các so sánh đã được thực hiện.

3.1. Thiết lập và thực hiện

3.1.1. Tập dữ liệu

Trong bài báo gốc, tác giả đã sử dụng dữ liệu thu thập được từ Microsoft Defender for Endpoint, tuy nhiên bộ dữ liệu này liên quan đến bảo mật thông tin khách hàng của Microsoft nên những người không có thẩm quyền sẽ không được tiếp cận. Do đó, nhóm chúng em đã tự tạo một phần lớn dữ liệu để sử dụng cho quá trình thực nghiệm này. Sau đây là các tập dữ liệu được sử dụng trong dự án:

- Dữ thiệu thô (Raw data): Đây chính là những dòng lệnh mà chưa được truyền vô bất kì tham số nào, trong quá trình tạo dữ liệu, tụi em đảm bảo rằng nó vẫn thể hiện được đầy đủ các cú pháp và trường hợp sử dụng dòng lệnh đó.

  Ví dụ: đối với lệnh "egsvr32 /s /n /u /i:http://example.com/file.sct scrobj.dll" thì ta sẽ có được dạng dữ liệu thô của nó là "regsvr32 /s /n /u /i".

- Dữ liệu được dán nhãn (Labeled data): Đây là những dòng lệnh đã được xác định là nó sẽ độc hại (malicious) hoặc lành tính (benign) nếu bị lợi dụng để thực hiện cho cuộc tấn công Living-Off-The-Land. Đối với những dòng lệnh độc hại, nhóm đã thu thập nó từ dự án (LOLBAS) [6], các lệnh này được công khai kèm với những hành động gây hại của nó lên hệ thống. Đối

với những dòng lệnh lành tính, nhóm đã tiến hành tự gán nhãn bằng cách xác định những hãy hưởng của lớp mà dòng lệnh đó thuộc về lên hệ thống, sau đó kiểm tra các tham số được truyền vào dòng lệnh đó có ảnh hưởng xấu đến hệ thống không, nếu đảm bảo là không thì dòng lệnh sẽ được dán nhãn lành tính.

Ví dụ: lệnh "ker.exe /start malware (malicious)" có thể xem là độc hại và đã được dán nhãn là malicious. Lệnh "CertOC.exe /list (benign)" có thể xem là lành tính và được dán nhãn là benign.

- Dữ liệu cần kiểm tra (Data need check): Đây là dòng lệnh chưa được dán nhãn và sẽ là đối tượng để ta thực nghiệm với mô hình sau khi xây dựng xong.

3.1.2. Phân đoạn (Tokenization)

Tokenization là quy trình cần thiết trong quá trị thực nghiệm này, vì mọi quy trình từ việc huấn luyện, đến kiểm tra và sử dụng đều trải qua việc tokenize. Khi trải qua bước này, các dòng lệnh sẽ được tách ra và mỗi từ trong dòng lệnh sẽ được coi là một token, đồng thời các số trong dòng lệnh cũng sẽ được chuyển thành một token chung là "number". Công việc này giúp ta làm sạch dữ liệu, loại bỏ những thành phần không cần thiết để không làm ảnh hưởng đến quá trình huấn luyện.

Ví dụ: Với dòng lệnh "eexec.exe http://11.11.11.11:8080/bypass.exe" sau khi tokenize sẽ trở thành "ieexec exe http number number number number number exe"

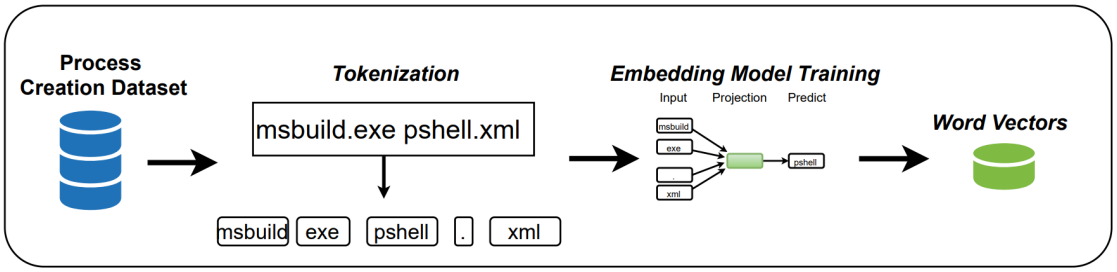
3.1.3. Mô hình nhúng từ (Word embedding)

1. Vai trò: Mô hình nhúng từ cho phép biểu diễn các từ (trong ngữ cảnh này là token) trong một không gian đa chiều, mỗi token sẽ được mô hình cung cấp một vector để thể hiện sự liên quan giữa các token với nhau trong ngữ

cảnh tổng quát. Các vector này cũng hỗ trợ đầu vào cho các bộ phân loại và máy học khi mà nó chỉ nhận dữ liệu số.

2. Thực hiện:

   - Chuẩn bị tài nguyên: Nhóm thực hiện huấn luyện mô hình fasttext trên Kali Linux bằng giao diện dòng lệnh. Trước tiên là tải thư viện fasttext về cùng với các gói yêu cầu để thư viện hoạt động, sau đó dữ liệu đầu vào chính là dữ liệu thô sẽ được tokenize được lưu trong tệp *rawCommand_ Tokenized.txt*



Hình 3.1: *Mô hình nhúng từ fasttext*
[7]

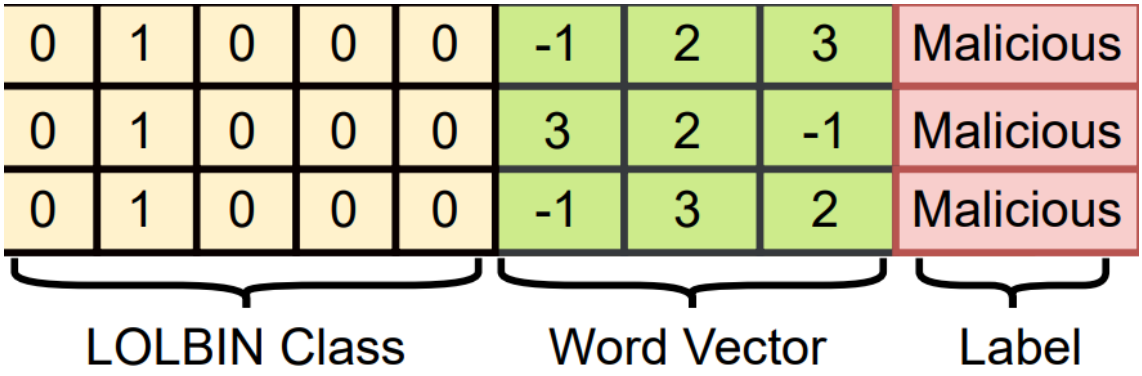
   - Tham số truyền vào và kết quả: Khi thực hiện huấn luyện, mô hình sẽ được truyền tham số 100 đối với dim (nghĩa là vector biểu diễn cho mỗi token sẽ có 100 chiều), minCount là 5 (tần số xuất hiện của token) và cuối cùng là window size với thông số là 5 (định nghĩa số token xung quanh token đang xét). Kết qủa, mô hình đã được huấn luyện sẽ được lưu vào *model_fasttext.bin*

3.1.4. Tạo điểm token

1. Vai trò: Điểm token là khoảng giá trị nằm trong khoảng từ 0 đến 1, càng về gần 1 thì nghĩa là token đó càng có xu hướng độc hại và ngược lại, trong một dòng lệnh độc hại thì sẽ có xu hướng chứa nhiều token có điểm số cao.

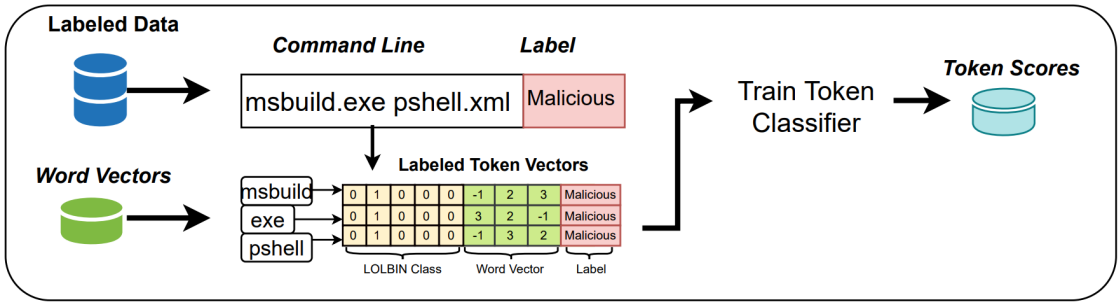
2. Thực hiện:

- Tạo Labeled Token Vector: Quá trình này sẽ cần 2 nguồn tài nguyên chính đó là bộ dữ liệu được dán nhãn và mô hình nhúng từ đã được huấn luyện ở bước trước. Đầu tiên, các dòng lệnh được dán nhãn sẽ được tokenize. Sau đó, với mỗi token, ta sẽ sử dụng mô hình nhúng từ để cung cấp các vector tương ứng cho nó, gọi là word vector. Bên cạnh đó, các lớp của dòng lệnh sẽ được biểu diễn one-hot-encoding, điều này cho phép biểu diễn trong các ngữ cảnh khác nhau hữu ít vì token có thể là độc hại hoặc lành tính tùy ngữ cảnh). Cuối cùng, để tạo nên Labeled Token Vector, ta sẽ nối biểu diễn one-hot cùng với word vector để có được vector tương ứng cho mỗi token.



Hình 3.2: *Labeled Token Vector*
[7]

- Huấn luyện bộ phân loại: Trong thực nghiệm này chúng em tiến hành thử nghiệm 4 loại mô hình là Random forest, light GBM, SVC (Support Vector Classifier) và XGBoost (XGB), trong đó sẽ có thêm 1 trường hợp là sử dụng ensemble learning để kết hợp 3 mô hình Random Forest, SVC và light GBM. Đầu vào cho các bộ phân loại này gồm có Labeled Token Vector và nhãn tương ứng của nó. Sau khi được huấn luyện, bộ phân loại này sẽ thực hiện công việc tính toán điểm cho từng token.



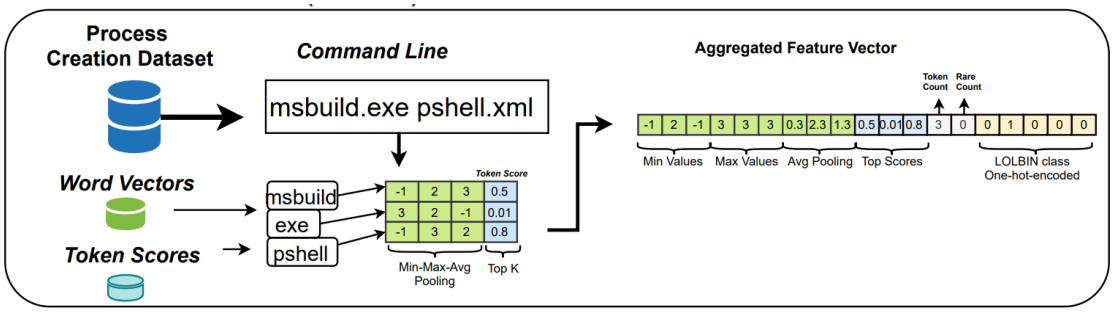
Hình 3.3: *Toàn bộ quy trình tạo ra điểm token*
[7]

3.1.5. Tạo vector đặc trưng tổng hợp

1. Vai trò: Vector đặc trưng tổng hợp (Aggregated Feature Vector) chính là vector thể hiện được đầy đủ các đặc trưng của dòng lệnh dựa vào các quy trình trước đó mà ta đã thực hiện, đây sẽ là lần chuyển đổi dữ liệu cuối cùng để hoàn thành việc huấn luyện mô hình đích.

2. Thực hiện:

   - Tính toán min-pooling, max-pooling và avg-pooling: dữ liệu truyền vô cho cả ba phép toán này chính là các word vector được nối với nhau có trong mỗi dòng lệnh. Tuy nhiên, với riêng avg-pooling, sẽ có thêm một tham số truyền vào là danh sách các điểm token của mỗi token có trong dòng lệnh, danh sách này sẽ đóng vai trò làm trọng số cho việc tính toán avg-pooling.



Hình 3.4: *Quá trình tạo ra vector đặc trưng tổng hợp*
[7]

- Gộp các thuộc đặc trưng: vector đặc trưng tổng hợp sẽ được tạo thành từ 7 thuộc tính lần lượt là: min-pooling, max-pooling, avg-pooling, top scores (chọn ra các điểm token cao nhất trong dòng lệnh, trong quá trình thực nghiệm thì nhóm chúng em chọn 2 điểm số cao nhất), số lượng token (đây là số lượng token có trong dòng lệnh tương ứng), token hiếm (đây là các token chưa xuất hiện trong dữ liệu để huấn luyện mô hình nhúng từ), cuối cùng sẽ là biểu diễn one-hot của lớp mà dòng lệnh đó thuộc về.

3.1.6. Mô hình quyết định

1. Huấn luyện Naive Bayes: từ vector đặc trưng tổng hợp đã tạo trước đó cùng với nhãn tương ứng của chúng, ta tiến hành huấn luyện mô hình Naive Bayes cụ thể là GaussianNB. Sau khi huấn luyện xong mô hình này, ta sẽ sử dụng nó để tính điểm uncertainty và anomaly.

2. Anomaly: Điểm số bất thường (Anomaly score) thể hiện mức độ không tương đồng của một mẫu so với các mẫu khác trong cùng một lớp. Điểm số này cho biết rằng mẫu đó được xếp vào một lớp nhất định, nhưng các thuộc tính và giá trị của mẫu đó lại có sự ít tương đồng so với các mẫu khác trong cùng một lớp.

$$A(n) = -\log P(\vec{x}|\text{class } c) = -\sum_{j=1}^{d} \log P(x_j|\text{class } c)$$

Trong đó:

- $A(n)$ là anomaly score của mẫu $\vec{x}$ được dự đoán thuộc vào class c.
- $P(\vec{x}|\text{class } c)$ là xác suất hậu nghiệm (posterior probability) rằng mẫu $\vec{x}$ thuộc vào class c, được dự đoán bởi bộ phân loại.

- $P(x_j|\text{class } c)$ là xác suất hậu nghiệm (posterior probability) rằng thuộc tính x_j của mẫu $\vec{x}$ thuộc vào class c.
- d là số lượng thuộc tính của mẫu $\vec{x}$.

3. Uncertainty[8]: Điểm số không chắc chắn (Uncertainty score) thể hiện sự không chắc chắn của một mẫu khi mà mô hình đã phân loại mẫu đó thuộc về một lớp, tuy nhiên các thuộc tính và giá trị của mẫu đó cũng có sự tương đồng với các mẫu thuộc lớp còn lại.

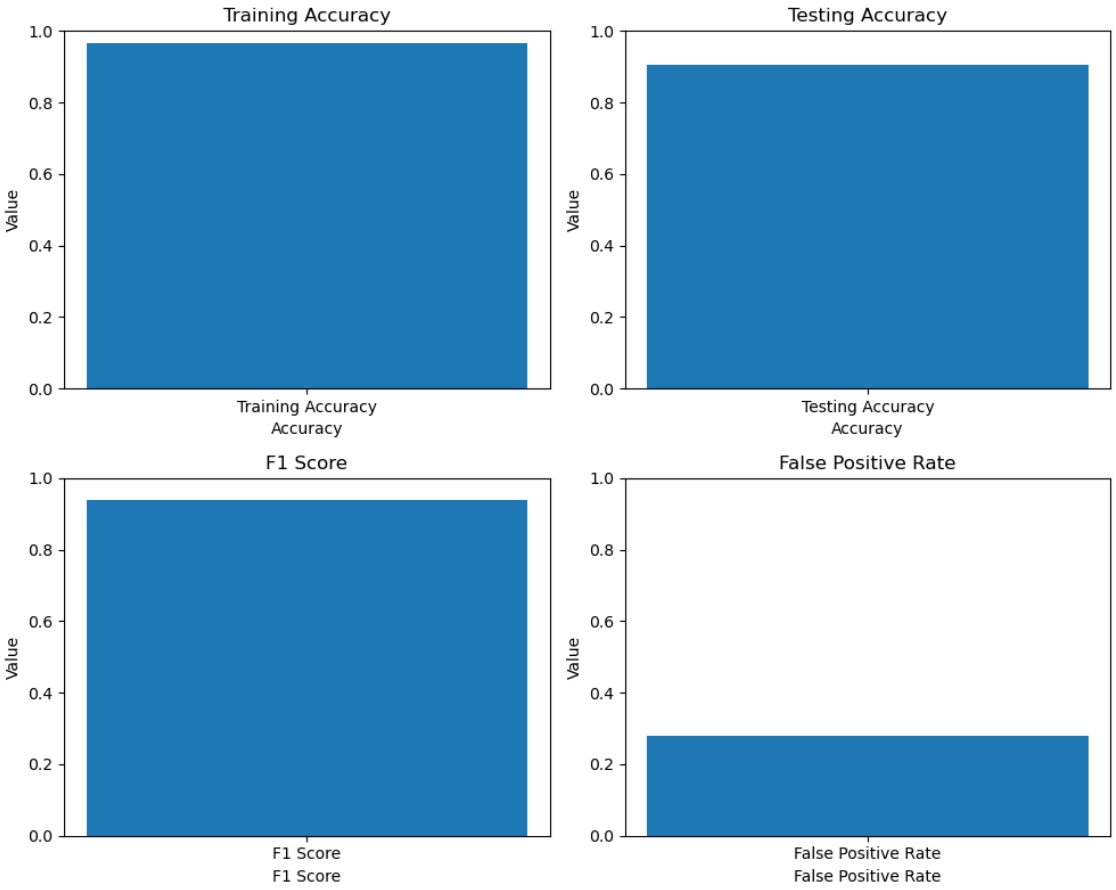
$$U(n) = -\min_{i,i\neq j} |P(i|\vec{x}_n) - P(j|\vec{x}_n)|$$

Trong đó:

- U(n) là uncertainty score của mẫu $\vec{x}_n$.
- $P(i|\vec{x}_n)$ là xác suất hậu nghiệm (posterior probability) rằng mẫu $\vec{x}_n$ thuộc vào class i.
- $P(j|\vec{x}_n)$ là xác suất hậu nghiệm (posterior probability) rằng mẫu $\vec{x}_n$ thuộc vào class j.
- $i = \text{argmax}_k P(k|\vec{x}_n)$ là class i mà mẫu $\vec{x}_n$ được dự đoán thuộc vào với xác suất cao nhất.

3.2. Đánh giá thực nghiệm

Trong mục này, chúng em sẽ thực hiện đánh giá đối với các bộ phân loại trong quá trình thực nghiệm.



Hình 3.5: *Các điểm số đánh giá mô hình Random Forest*

3.2.1. Các bộ phân loại để tính điểm token

3.2.1.1. Random Forest

Mô hình Random Forest cho kết quả khả quan trong quá trình huấn luyện, với độ chính xác đạt khoảng 96.7%. Điều này có nghĩa là mô hình có khả năng học hỏi từ dữ liệu huấn luyện và phân loại các mẫu trong tập dữ liệu một cách chính xác cao. Và trong quá trình áp dụng mô hình với tập dữ liệu kiểm tra, độ chính xác đạt khoảng 90%, đây cũng là một con số cao.

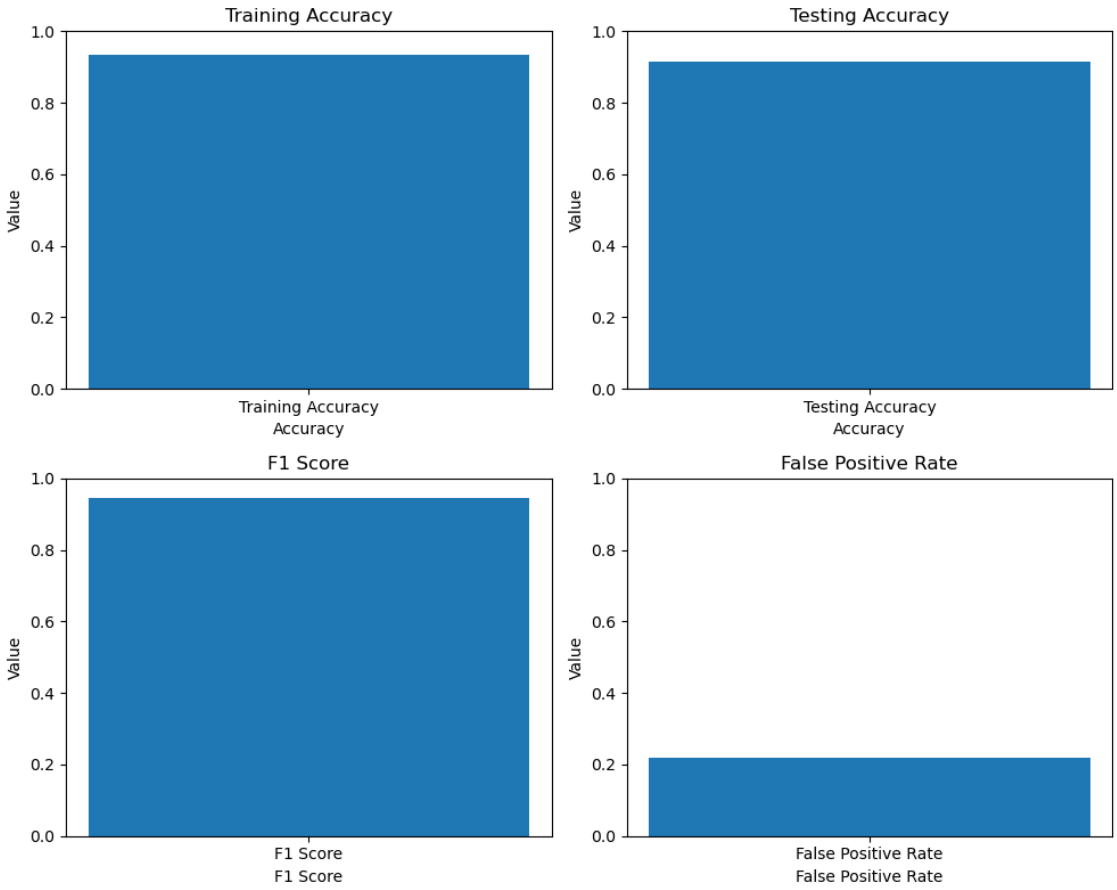
F1 score của mô hình là 0.93811, đây là một chỉ số kết hợp giữa độ chính xác và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình Random Forest có khả năng phân loại tốt các lớp.

False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.28, có nghĩa là các trường hợp bị phân loại là độc hại nhưng thực tế thì không phải. Con số này tương đối khá cao, điều này có nghĩa là mô hình cần cải thiện trong tương lai.

Tổng thể, mô hình Random Forest cho kết quả tương đối khả quan trong việc phân loại các mẫu. Tuy nhiên, do được huấn luyện trên tập dữ liệu nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên một tập dữ liệu thực tế lớn hơn.

3.2.1.2. SVC (Support Vector Classifier)

Mô hình SVC cho kết quả khả quan trong quá trình huấn luyện, với độ chính xác đạt khoảng 93.43%. Điều này có nghĩa là mô hình có khả năng học hỏi từ dữ liệu huấn luyện và phân loại các mẫu trong tập dữ liệu một cách chính xác cao. Và trong quá trình áp dụng mô hình với tập dữ liệu kiểm tra, độ chính xác đạt khoảng 91.45%, đây là một con số tương đối cao. Khi so sánh hai chỉ số về độ chính xác trong quá trình huấn luyện và kiểm tra, ta thấy kết quả thu được là tương đối bằng nhau, nên mô hình này gọi là goodfitting.



Hình 3.6: *Các điểm số đánh giá mô hình Support Vector Classifier*

F1 score của mô hình là 0.943, đây là một chỉ số kết hợp giữa độ chính xác và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình SVC có khả năng phân loại tốt các lớp.

False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.22, có nghĩa là các trường hợp bị phân loại là độc hại nhưng thực tế thì không phải. Con số này tương đối khá cao so với các mô hình còn lại.

Tổng thể, mô hình LightGBM cho kết quả tương đối khả quan trong việc phân loại các mẫu. Tuy nhiên, do được huấn luyện trên tập dữ liệu nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên một tập dữ liệu thực tế lớn hơn.

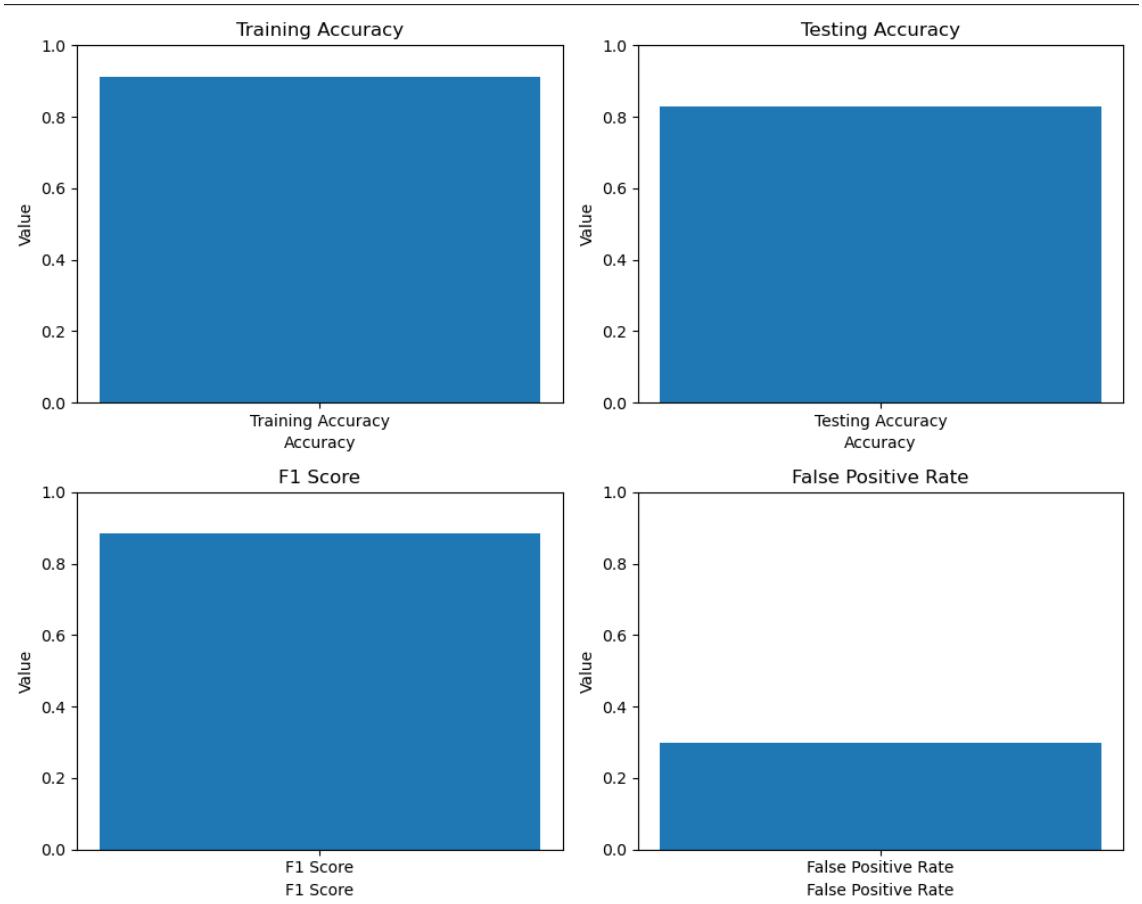
3.2.1.3. Light GBM

Mô hình LightGBM cho kết quả khả quan trong quá trình huấn luyện, với độ chính xác đạt khoảng 91.03%. Điều này có nghĩa là mô hình có khả năng học hỏi từ dữ liệu huấn luyện và phân loại các mẫu trong tập dữ liệu một cách chính xác cao. Và trong quá trình áp dụng mô hình với tập dữ liệu kiểm tra, độ chính xác đạt khoảng 82.91%, đây là một con số tương đối tốt.

F1 score của mô hình là 0.884, đây là một chỉ số kết hợp giữa độ chính xác và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình LightGBM có khả năng phân loại tốt các lớp.

False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.3, có nghĩa là các trường hợp bị phân loại là độc hại nhưng thực tế thì không phải. Con số này tương đối khá cao, điều này có nghĩa là mô hình cần cải thiện trong tương lai.

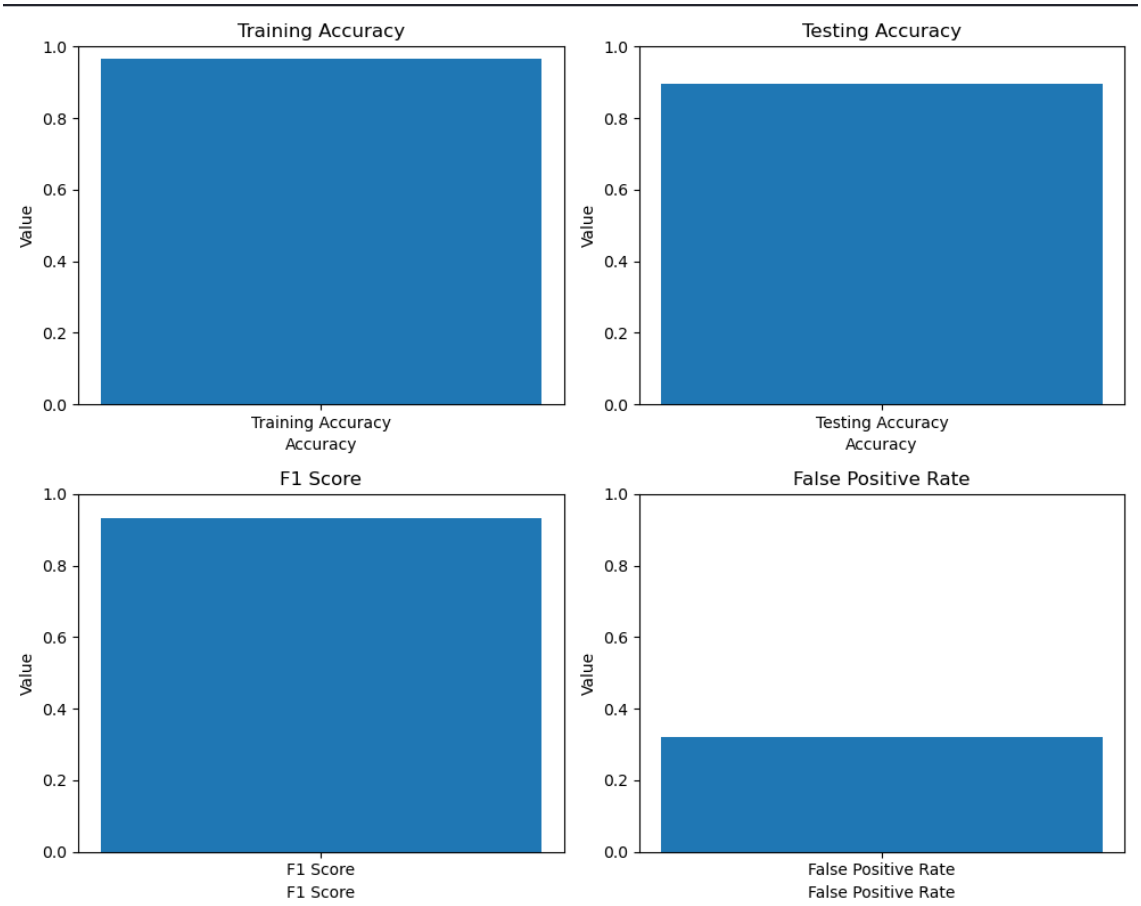
Tổng thể, mô hình LightGBM cho kết quả tương đối khả quan trong việc phân loại các mẫu. So sánh với mô hình Random Forest, thì mô hình LightGBM cho kết quả không cao bằng, nhưng đây cũng là một con số tương đối tốt trong việc huấn luyện và phân loại. Tuy nhiên, do được huấn luyện trên tập dữ liệu



Hình 3.7: *Các điểm số đánh giá mô hình Light Gradient Boosting Machine*

nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên một tập dữ liệu thực tế lớn hơn.

3.2.1.4. Ensemble model



Hình 3.8: *Các điểm số đánh giá mô hình Ensemble*

Mô hình Ensemble cho kết quả tốt trong quá trình huấn luyện, với độ chính xác đạt khoảng 96.46%. Điều này có nghĩa là mô hình có khả năng học hỏi từ dữ liệu huấn luyện và phân loại các mẫu trong tập dữ liệu một cách chính xác cao. Và trong quá trình áp dụng mô hình với tập dữ liệu kiểm tra, độ chính xác đạt khoảng 89.44%, đây là một con số tương đối tốt.

F1 score của mô hình là 0.932, đây là một chỉ số kết hợp giữa độ chính xác

và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình Ensemble có khả năng phân loại tốt các lớp.

False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.32, có nghĩa là các trường hợp bị phân loại là độc hại nhưng thực tế thì không phải. Con số này tương đối khá cao, điều này có nghĩa là mô hình cần cải thiện trong tương lai.

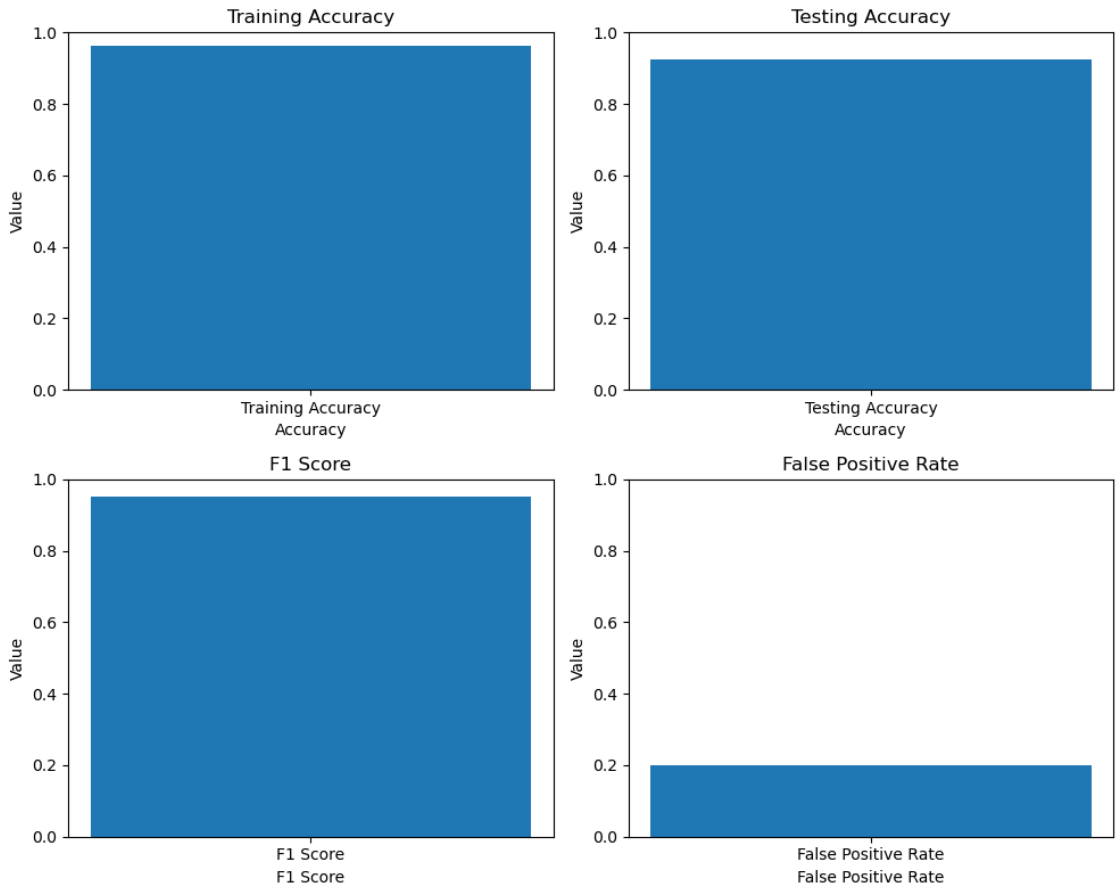
Tổng thể, mô hình Ensemble cho kết quả tương đối khả quan trong việc phân loại các mẫu. So sánh với mô hình Random Forest. Tuy nhiên, do được huấn luyện trên tập dữ liệu nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên một tập dữ liệu thực tế lớn hơn. Việc sử dụng mô hình Ensemble sự kết hợp giữa 3 mô hinh Random Forest, LightGBM, và SVC để trong tương lai, khi dữ liệu thay đổi thì không loại trừ khả năng bị overfitting.

3.2.1.5. XGBoost (XGB)

Mô hình XGB cho kết quả tốt trong quá trình huấn luyện, với độ chính xác đạt khoảng 96.33%. Điều này có nghĩa là mô hình có khả năng học hỏi từ dữ liệu huấn luyện và phân loại các mẫu trong tập dữ liệu một cách chính xác cao. Và trong quá trình áp dụng mô hình với tập dữ liệu kiểm tra, độ chính xác đạt khoảng 92.46%, đây là một con số tương đối tốt. Khi so sánh hai chỉ số về độ chính xác trong quá trình huấn luyện và kiểm tra, ta thấy kết quả thu được là tương đối bằng nhau, nên mô hình này gọi là goodfitting.

F1 score của mô hình là 0.95, đây là một chỉ số kết hợp giữa độ chính xác và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình XGB có khả năng phân loại tốt các lớp.

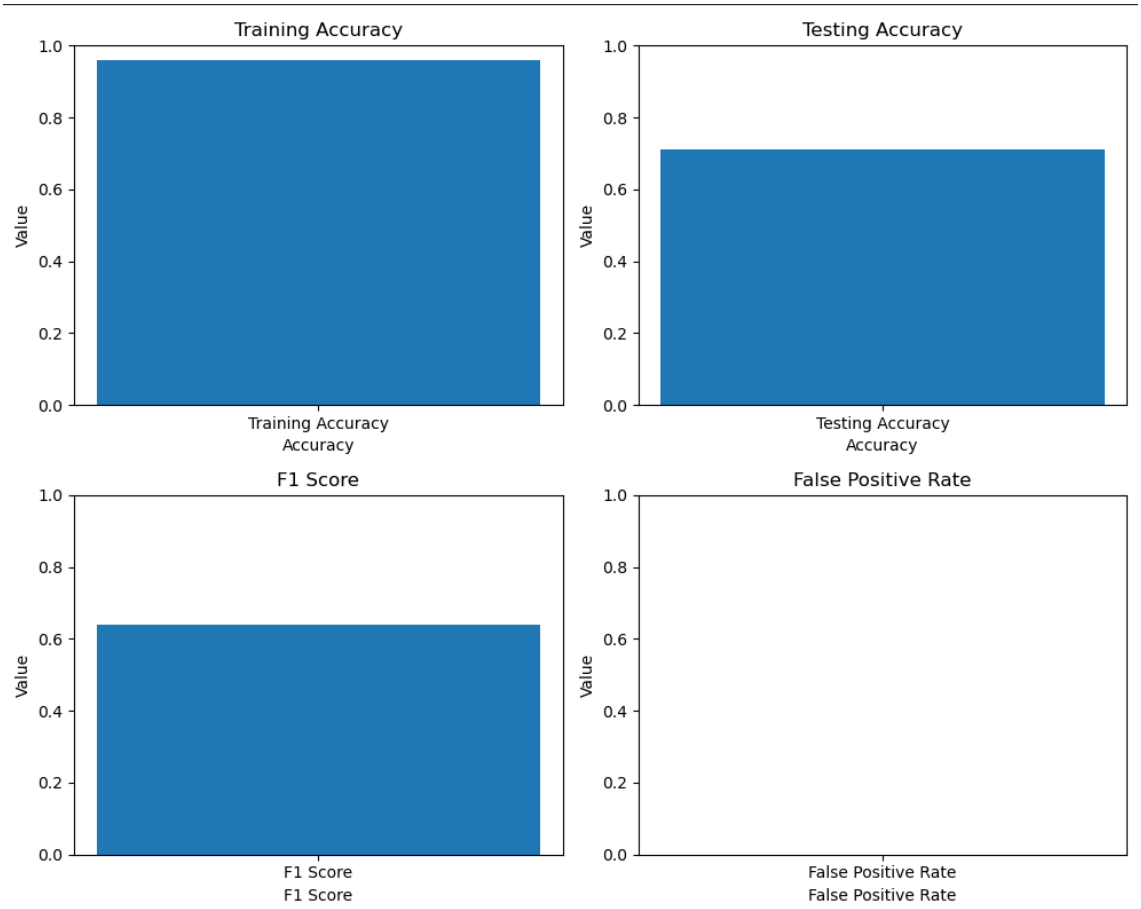
False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.2, có nghĩa là các trường hợp bị phân loại là độc hại nhưng thực tế thì không phải. Con số này tương đối khá cao, điều này có nghĩa là mô hình cần cải thiện trong tương lai.



Hình 3.9: *Các điểm số đánh giá mô hình XGBoost*

Tổng thể, mô hình XGB cho kết quả tốt nhất trong các mẫu trên. Tuy nhiên, do được huấn luyện trên tập dữ liệu nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên một tập dữ liệu thực tế lớn hơn.

3.2.2. Đánh giá mô hình tổng



Hình 3.10: *Các điểm số đánh giá mô hình Naive Bayes*

Mô hình Naive Bayes đã cho kết quả đáng chú ý trong quá trình huấn luyện, với độ chính xác đạt khoảng 95.9%. Điều này cho thấy mô hình có khả năng học từ dữ liệu huấn luyện và phân loại các mẫu trong tập này một cách chính xác cao. Tuy nhiên, khi áp dụng mô hình lên tập dữ liệu kiểm tra, độ chính xác giảm xuống khoảng 70.9%. Điều này cho thấy mô hình có khả năng không tổng quát hoá tốt từ dữ liệu huấn luyện sang dữ liệu mới. Cần lưu ý rằng mô hình

chỉ được huấn luyện trên tập dữ liệu nhỏ, có thể không đủ đại diện cho toàn bộ các trường hợp có thể xảy ra.

F1 score của mô hình là 0.6399, đây là một chỉ số kết hợp giữa độ chính xác và độ phủ của mô hình trong việc phân loại các lớp. Điểm số này cho thấy mô hình Naive Bayes có khả năng phân loại tốt các lớp, nhưng còn một số sai sót trong việc phủ sót và chính xác các mẫu.

False positives rate (Tỷ lệ phân loại sai dương tính) của mô hình là 0.0, tức là không có trường hợp nào bị phân loại sai thành dương tính trong tập dữ liệu kiểm tra. Điều này cho thấy mô hình có khả năng phân loại các mẫu âm tính một cách chính xác.

Tổng thể, mô hình Naive Bayes đã cho kết quả tương đối tốt trong việc phân loại các mẫu. Tuy nhiên, do được huấn luyện trên tập dữ liệu nhỏ và có thể không đại diện đầy đủ, nên cần xem xét kỹ hơn và cải thiện mô hình để đạt được hiệu suất phân loại tốt hơn trên dữ liệu thực tế.

3.2.3. Đánh giá quyết định của mô hình

Vì một số đặc trưng quan trong của mô hình Naive Bayes, cụ thể là độc lập tuyến tính, nên tụi em đã thống nhất đây chính là mô hình chính của thực nghiệm

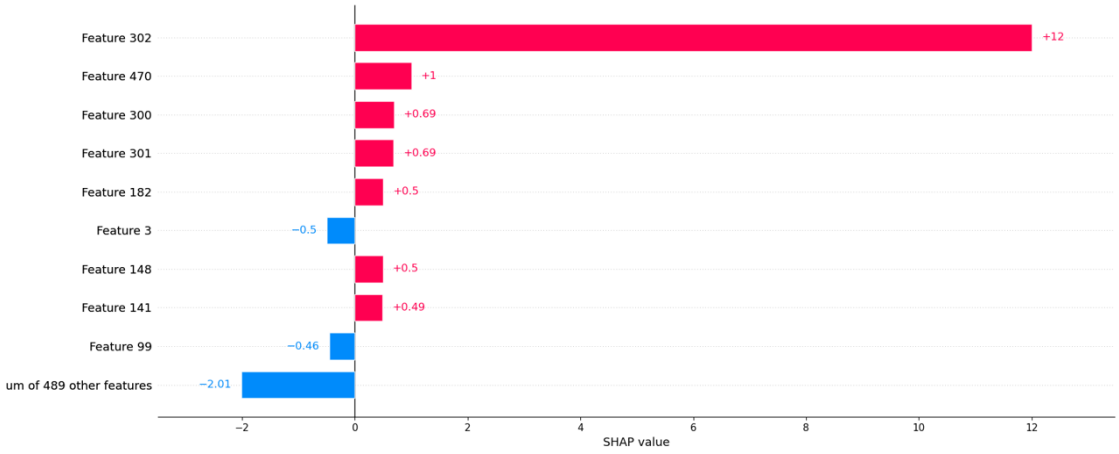
Trước khi tiến hành phân tích quyết định của mô hình, ta sẽ định nghĩa lại các đặc trưng của dữ liệu đầu vào. Đầu vào của mô hình Naive Bayes sẽ là vector đặc trưng tổng hợp gồm 7 thành phần và tổng cổng có 498 chiều nên ta sẽ thống nhất như sau:

- Chiều từ 1 đến 100 : min-pooling
- Chiều từ 101 đến 200 : max-pooling
- Chiều từ 201 đến 300 : avg-pooling
- chiều từ 301 đến 302 : top scores

- Chiều thứ 303 : number token
- Chiều thứ 304 : rare token
- Chiều thứ 305 đến 498 : one-hot (class vector)

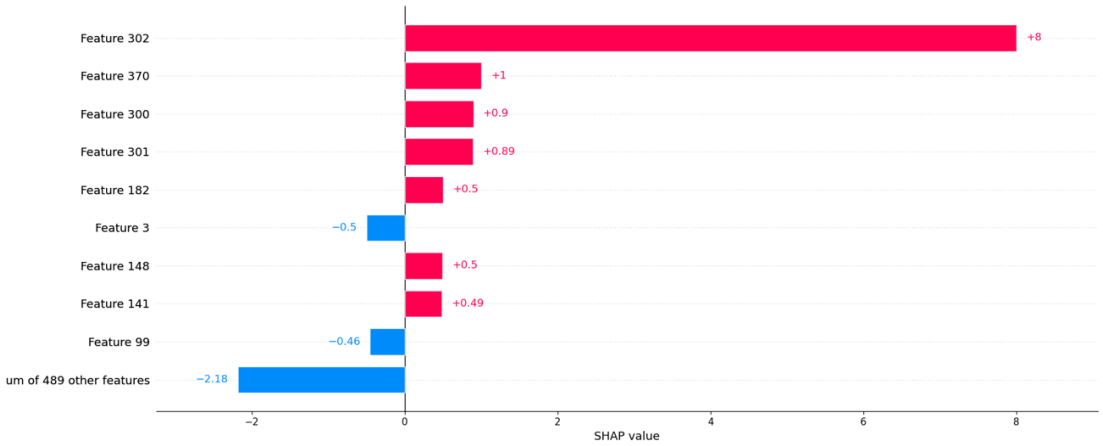
3.2.3.1. SHAP framework

SHAP tập trung vào việc phân tích quyết định của mô hình dự đoán bằng cách xác định mức đóng góp của từng đặc trưng vào dự đoán. Nó giúp ta hiểu rõ hơn về tầm quan trọng và ảnh hưởng của mỗi đặc trưng trong việc ra quyết định của mô hình. SHAP sẽ tính toán giá trị Shapley để xác định sự đóng góp của từng đặc trưng bằng cách xem xét tất cả các khả năng kết hợp của các đặc trưng trong các tập con của dữ liệu.

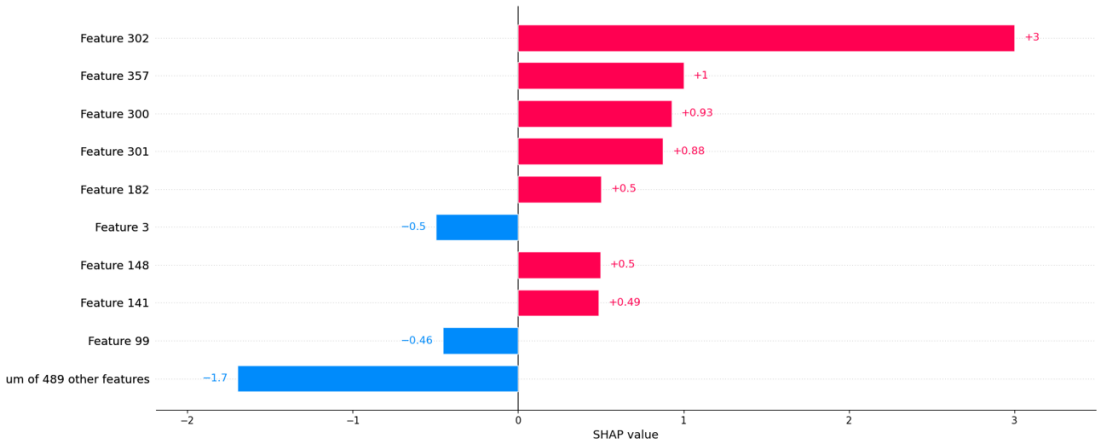


Hình 3.11: *Mẫu 1*

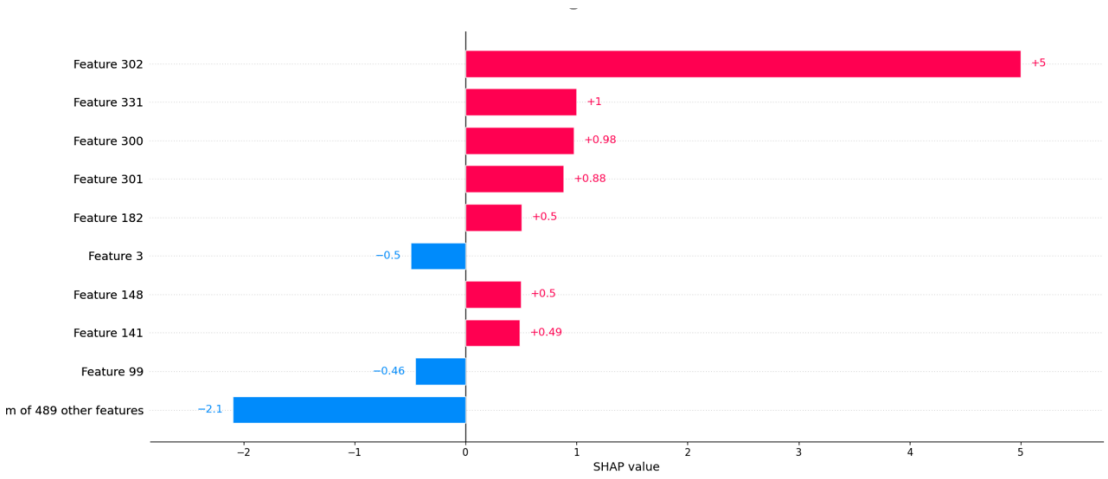
Nhận xét: Trong 9 thuộc tính gây ảnh hưởng lớn nhất lên mô hình có thì cả 8 thuộc tính vẫn luôn duy trình được sự ảnh hưởng cố định tuy nhiên khi xét chung về 7 thành phần của của vector đặc trưng tổng hợp thì có thể thấy sự ảnh hưởng của các thành phần này là gần như không đổi. "Top scores" được xem là có ảnh hưởng nhất, điều này là dễ hiểu khi nó mang các điểm số lớn nhất của token có trong dòng lệnh, thứ mà quyết định xem token đó là độc hại hay lành



Hình 3.12: *Mẫu 2*



Hình 3.13: *Mẫu 3*

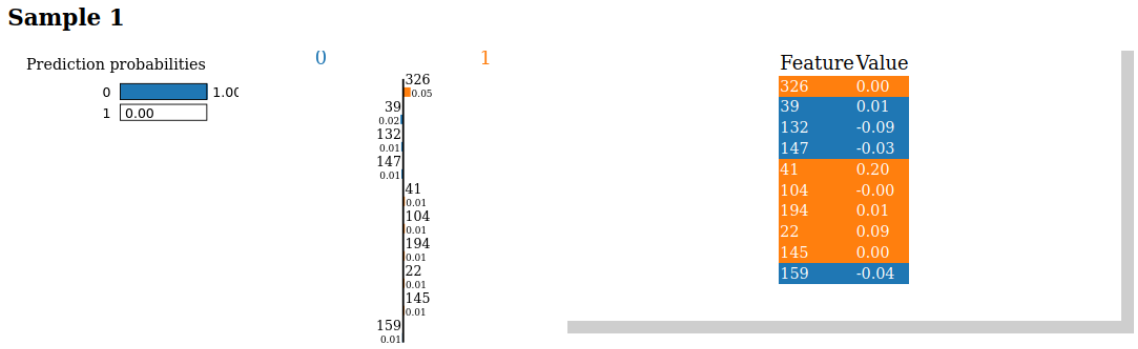


Hình 3.14: *Mẫu 4*

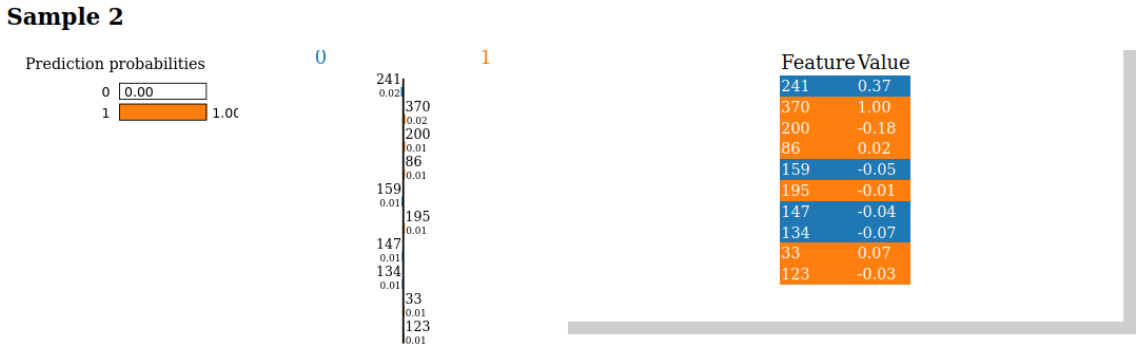
tính. Để chắc chắn hơn về phân tích của SHAP, nhóm em của đã kiểm tra chiều của vector đặc trưng tổng hợp gây ảnh hưởng lớn lên mô hình và thấy rằng sai số giữa các mẫu ở những chiều này là rất nhỏ khi được xếp vào chung một lớp.

3.2.3.2. Lime framework

Lime tập trung vào việc phân tích dữ liệu đầu vào và sự ảnh hưởng của nó đến quyết định của mô hình. Nó tạo ra các giải thích cục bộ bằng cách xây dựng một mô hình giải thích đơn giản gần đúng cho vùng lân cận của điểm dữ liệu quan tâm. LIME tạo ra các mẫu dữ liệu được biến đổi ngẫu nhiên từ dữ liệu gốc và sử dụng chúng để xây dựng mô hình giải thích.

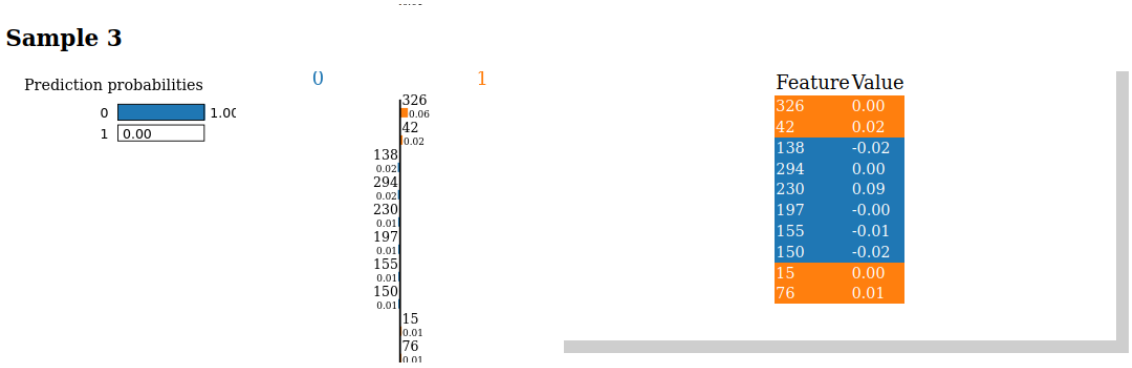


Hình 3.15: *Mẫu 1*

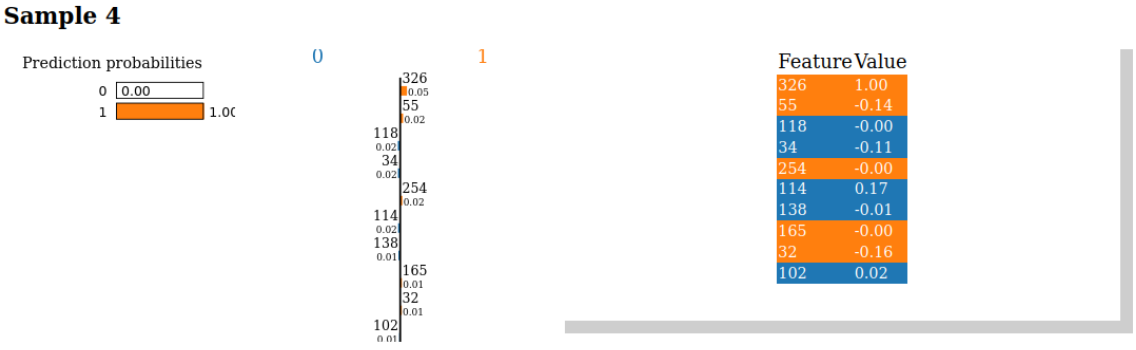


Hình 3.16: *Mẫu 2*

Nhận xét: Sự chênh lệch giữa đa số các thuộc tính đầu vào ảnh hưởng với mô hình có thể thấy là không đáng kể, thậm chí là như nhau điều này em nghĩ có thể được lý giải qua việc mô hình nhận các giá trị này là Naive Bayes, nó có



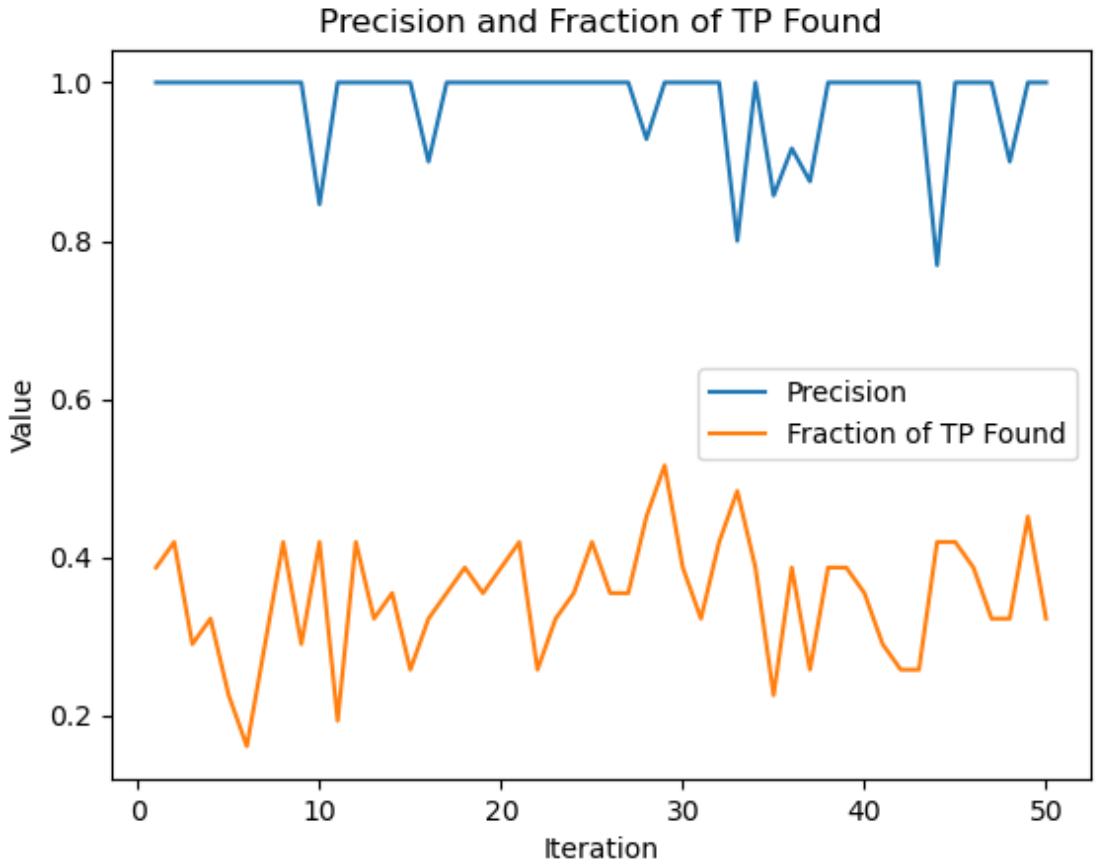
Hình 3.17: *Mẫu 3*



Hình 3.18: *Mẫu 4*

tính chất gọi là độc lập tuyến tính, điều này giúp mô hình xem các chiều của vector đầu vào là "bình đẳng" vì, mô hình xem các chiều là các giá trị độc lập và không liên quan đến nhau.

3.2.4. Học chủ động (Active learning)



Hình 3.19: *Precision và tỷ lệ số lượng True Positive tìm thấy*

Thực hiện active learning, mô hình được chạy qua 50 vòng lặp, sau mỗi vòng lặp mô hình sẽ chọn ra các mẫu có điểm uncertainty và anomaly cao để dán nhãn tay và học mẫu đó để tiếp tục tiến hành các vòng lặp sau.

Tổng quan mô hình hiện tại cho ta thấy điểm precision cao, hầu hết qua các mỗi vòng lặp đều đặt ngẫn ngưỡng 1, vẫn có trường hợp điểm này bị hạ xuống khoảng từ 0,75 đế 0.9, tuy nhiên xét về kết quả trung bình thì nhận định được mô hình đạt được khoảng 0.9 điểm precision. Điều này cho thấy khả năng dự

đoán chính xác các trường hợp dương tính (True Positive) của mô hình là tương đối cao. Tuy nhiên, chúng ta cũng cần chú ý đến tỷ lệ số lượng True Positive tìm thấy (fraction of TP found) của mô hình. Tỷ lệ này chỉ ra khả năng của mô hình trong việc phát hiện các trường hợp dương tính thực tế, trong trường hợp này tỷ lệ số lượng True Positive tìm thấy thấp, giao động từ 0.1 đến khoảng gần 0.6 và có sự biến thiên rõ rệt qua mỗi vòng lặp. Mặc dù precision cao, có thể mô hình đang bỏ sót một số trường hợp dương tính.

Giải thích kết quả của mô hình. Trong bài báo gốc, tác giả thực nghiệm với từng class (Bitsadmin, certutil,...) cụ thể thay vì toàn bộ class như nhóm thực hiện, điều này khiến cho mô hình không nắm rõ toàn bộ ngữ cảnh và phân định với từng class được nên dẫn đến kết quả không khả quan. Lý do thực hiện toàn bộ class cùng lúc thay vì từng class cụ thể là vì, với mỗi class nhóm chúng em chỉ kiếm được nhiều nhất là 5 đến 6 dòng lệnh, trong khi đó muốn thấy được sự hiểu quả của acctive learning thì cần lặp số lượng vòng lặp rất nhiều lần. Vì vậy bọn em thay vì kiếm nhiều lệnh hơn cho mỗi class (điều này là khó vì hiểu được ngữ cảnh của một số lượng lớn lệnh thuộc cùng một class đòi hỏi nhiều thời gian) thì bọn em sẽ thực hiện kiếm nhiều class hơn và mỗi class sẽ có khoảng từ 3 đến 5 dòng lệnh, tổng cộng có khoảng 40 class được huấn luyện cho mô hình.

CHƯƠNG 4. KẾT LUẬN

Ở chương này, chúng em đưa ra những kết luận về nghiên cứu, những hạn chế, và đồng thời đưa ra hướng cải thiện và phát triển.

4.1. Kết luận

Mô hình không cho thấy sự hội tụ của điểm precision và tỷ lệ số lượng True Positive tìm thấy, sự cách biết giữa hai điểm này là lớn. Tuy nhiên khả năng tổng quát hóa của mô hình có thể bị giới hạn vì số lượng dữ liệu là rất nhỏ trong khi số lượng dòng lệnh thuộc các lớp là rất lớn nên có thể ảnh hưởng đến kết quả của bộ phân loại. Bên cạnh đó, có một số vấn đề không được tác giả làm rõ trong bài báo gốc cũng có thể ảnh hưởng nhỏ đến mô hình thực hiện của tụi em tuy nhiên chắc chắn sự ảnh hưởng là không lớn vì đã được xem xét lại kỹ về mặc logic, đồng thời mã nguồn được chia nhỏ nên có thể dễ dàng chỉnh sửa nếu cần sửa đổi.

4.2. Hướng phát triển

Hướng đi tiếp theo của đề tài có thể tập trung vào việc tăng cường khả năng phát hiện trong môi trường thực tế. Để làm điều này, nghiên cứu có thể tận dụng các kỹ thuật học tăng cường để tự động học các lệnh và hành vi mới của kẻ tấn công.

Một khía cạnh quan trọng là khả năng tìm hiểu tự động và cập nhật mô hình dựa trên dữ liệu realtime. Đây làm một yếu tố quan trọng để đảm bảo tính hiệu quả và linh hoạt khi đối mặt với các hình thức tấn công mới. Bằng cách liên tục thu thập và phân tích dữ liệu realtime, hệ thống có thể tự động học từ các hành

vi và lệnh mới của kẻ tấn công, từ đó nâng cao khả năng phát hiện và phản ứng.

Nghiên cứu có thể tập trung vào việc phát triển các thuật toán và mô hình học tăng cường, để giúp hệ thống tự động nhận biết và phân loại các lệnh và hành vi mới của kẻ tấn công. Việc sử dụng dữ liệu realtime để đào tạo và cập nhật mô hình có thể được thực hiện thông qua việc phát hiện và thu thập dữ liệu từ các cuộc tấn công thực tế hoặc sử dụng các phương pháp thu thập dữ liệu động và liên tục.

TÀI LIỆU THAM KHẢO

Tiếng Anh:

[1] DMLC, *XGBoost Repository*, N/A, URL: `https://github.com/dmlc/xgboost`.

[2] Leo Breiman (2001), "Random forests", *Machine Learning*, 45 (1), pp. 5–32.

[3] Corinna Cortes and Vladimir Vapnik (1995), "Support Vector Machine Classifiers", *Machine Learning*, 20 (3), pp. 273–297.

[4] Guolin Ke et al., "LightGBM: A highly efficient gradient boosting decision tree", in: *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[5] Irina Rish (2004), "The Naive Bayes classifier", *Machine learning*, 55 (2), pp. 141–166.

[6] LOLBAS Project, *Living Off The Land Binaries and Scripts (and also Libraries)*, 2021, URL: `https://lolbas-project.github.io/`.

[7] Tayfun Ongun et al., "Living-off-the-land command detection using active learning", in: *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*, 2021, pp. 442–455.

[8] David D. Lewis and William A. Gale, "A sequential algorithm for training text classifiers", in: *SIGIR'94*, Springer, 1994, pp. 3–12.